

AI-03004 PostgreSQL Information

Owner	© Rigel Arcayan
Tags	Project
Created time	@September 16, 2025 8:03 PM

What is PostgreSQL? What is the history of the database creation and development? Create a comprehensive guide in the installation, deployment, management. List all the commands used in the Ubuntu server environment. Create a guide on how to perform basic database creation, user creation, passwords, logins, reset account, drop db etc. create a guide on how to install and configure phasing , it's prerequisites and needed dependencies.

PostgreSQL, or Postgres, is a powerful, open-source, object-relational database management system (ORDBMS) known for its reliability, data integrity, and extensibility. It's often used for mission-critical applications and is highly compliant with the SQL standard.

History of PostgreSQL

PostgreSQL's origins trace back to a project at the University of California, Berkeley.

- **1986:** The project, initially called **POSTGRES**, was led by Professor Michael Stonebraker. It was a successor to the earlier Ingres database, also developed at Berkeley. The goal of POSTGRES was to address the limitations of traditional relational databases by introducing features like object-oriented concepts and extensibility.
- **1996:** The project was renamed **PostgreSQL** to reflect its full support for the SQL language. A community of developers took over the project after its academic life at Berkeley concluded.

- **Present:** The PostgreSQL Global Development Group, a dedicated community of contributors, continues to develop and release new versions of the database. PostgreSQL has grown to become one of the most popular and respected open-source databases in the world, valued for its stability and advanced features.
-

Installation and Management on Ubuntu

This guide will walk you through the process of installing and performing basic management tasks for PostgreSQL on an Ubuntu server. All commands are run from the terminal.

Installation and Deployment

1. **Update your system's package list:**Bash

```
sudo apt update
```

2. Install the PostgreSQL package:Bash

This command installs the server, client, and a few extensions.

```
sudo apt install postgresql postgresql-contrib
```

3. Check the status of the PostgreSQL service:Bash

The service starts automatically after installation.

```
sudo systemctl status postgresql
```

4. **Connect to the PostgreSQL shell (psql) as the default `postgres` user:**Bash

```
sudo -u postgres psql
```

You'll be dropped into the `psql` shell, indicated by the prompt `postgres=#`. To exit, type `\q` and press Enter.

Basic Database and User Management

The `postgres` user is a superuser with full administrative rights. You should perform administrative tasks from this account.

- **Creating a new database:**BashSQL

```
sudo -u postgres createdb your_new_database_name
```

Alternatively, from the `psql` shell:

```
CREATE DATABASE your_new_database_name;
```

- **Creating a new user (role):**BashSQL

You can create a user from the command line or from within the `psql` shell.

```
# From the command line, this will prompt for a password  
sudo -u postgres createuser --interactive
```

- - From the `psql` shell

```
CREATE USER your_new_user_name WITH PASSWORD 'your_secure_password';
```

- **Granting privileges to a user on a database:**SQL

This command gives `your_new_user_name` full privileges on `your_new_database_name`.

```
GRANT ALL PRIVILEGES ON DATABASE your_new_database_name TO your_new_user_name;
```

- **Changing a user's password:**SQL

```
ALTER USER your_user_name WITH PASSWORD 'new_secure_password';
```

- **Dropping a user:**BashSQL

```
sudo -u postgres dropuser your_user_name
```

Alternatively, from the `psql` shell:

```
DROP USER your_user_name;
```

- **Dropping a database:**BashSQL

Make sure no one is connected to the database before dropping it.

```
sudo -u postgres dropdb your_database_name
```

Alternatively, from the `psql` shell:

```
DROP DATABASE your_database_name;
```

Installing and Configuring PgHero

PgHero is a performance dashboard for PostgreSQL that provides an easy-to-read interface for monitoring your database. It's built as a Ruby on Rails application, but it can be installed and run standalone.

Prerequisites and Dependencies

- **Ruby and Bundler:** PgHero is a Ruby gem, so you'll need **Ruby** and **Bundler** to install its dependencies.
- **PostgreSQL:** You must have a running PostgreSQL instance.
- **`pg_stat_statements` extension:** This is a crucial extension for PgHero to gather detailed query statistics. You'll need to enable it in your PostgreSQL configuration.

Installation and Configuration

1. Install Ruby and other dependencies: Bash

```
sudo apt install ruby-full build-essential  
sudo gem install bundler
```

2. Install the PgHero gem: Bash

```
gem install pghero
```

3. Enable `pg_stat_statements` :

- Open your `postgresql.conf` file. The location can vary, but it's typically in `/etc/postgresql/your_version/main/postgresql.conf`. Bash

```
sudo nano /etc/postgresql/16/main/postgresql.conf
```

- Find the `shared_preload_libraries` line and add `'pg_stat_statements'`. If it's commented out, uncomment it. Code snippet

```
shared_preload_libraries = 'pg_stat_statements'
```

- Save and close the file.
- Restart the PostgreSQL service for the changes to take effect. Bash

```
sudo systemctl restart postgresql
```

- Connect to a database in `psql` and create the extension. You only need to do this once per database you want to monitor. SQL

```
CREATE EXTENSION pg_stat_statements;
```

4. Configure and run PgHero:

- Create a file to store your database connection string, for example, `pghero.env`.
- Add your database connection URL to the file. The format is `postgres://user:password@host:port/database`. Code snippet

```
DATABASE_URL=postgres://your_user:your_password@localhost:5432/your_database
```

- Run PgHero using the `pghero` command with your environment file. Bash

```
pghero --database-url=postgres://your_user:your_password@localhost:5432/your_database
```

- PgHero will now run a web server. You can access the dashboard in your browser at `http://localhost:8080` (or a different port if specified).

pgAdmin is a popular, open-source graphical user interface (GUI) for managing PostgreSQL databases. Installing it on an Ubuntu server allows you to manage your databases through a web browser. Here are the steps to install and set up pgAdmin in web mode on Ubuntu.

Step 1: Add the pgAdmin Repository

Since pgAdmin isn't included in the default Ubuntu repositories, you need to add its official repository.

First, install `curl` to download the repository key:

Bash

```
sudo apt update  
sudo apt install curl ca-certificates -y
```

Next, add the public key for the repository to your system's keyring:

Bash

```
curl -fsSL https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o  
/etc/apt/trusted.gpg.d/pgadmin.gpg
```

Finally, add the repository to your `sources.list.d` directory and update your package list:

Bash

```
sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/${lsb_release -cs} pgadmin4 main" >  
/etc/apt/sources.list.d/pgadmin4.list'  
sudo apt update
```

Step 2: Install pgAdmin 4

Now that the repository is configured, you can install the pgAdmin package. For a web-based interface, install `pgadmin4-web`.

Bash

```
sudo apt install pgadmin4-web
```

This will also install the Apache web server and other necessary dependencies.

Step 3: Configure pgAdmin Web Mode

After installation, you need to configure the web server and create a login for the pgAdmin web interface. Run the setup script provided by the package:

Bash

```
sudo /usr/pgadmin4/bin/setup-web.sh
```

This script will prompt you to enter an **email address** and a **password** that will be used to log in to the pgAdmin web interface.

Step 4: Access pgAdmin

Once the setup script finishes, you can access the pgAdmin web interface from any browser on your network. Navigate to your server's IP address or domain followed by `/pgadmin4`.

`http://<your-server-ip>/pgadmin4`

You will be greeted with the login screen. Use the email and password you set in the previous step to log in.

After logging in, you will be on the main dashboard. To connect to a PostgreSQL server, click on "Add New Server" in the dashboard and provide the connection details, such as the host, port, username, and password for your database.

Step 5: Connect to a PostgreSQL Server

To connect to your PostgreSQL instance, you'll need the following information:

- **Host:** For a local server, this will be `localhost` or `127.0.0.1`. For a remote server, use its IP address.
- **Port:** The default PostgreSQL port is **5432**.
- **Username:** The database user you want to connect with (e.g., `postgres` or a user you created).
- **Password:** The password for that user.

After entering these details under the "Connection" tab in the "Create - Server" window, click **Save** to add the server to pgAdmin.

Here is a video guide on how to install PostgreSQL and pgAdmin on an Ubuntu server.

How to Install PostgreSQL and PGAdmin on Ubuntu